

```

options( java.parameters = "-Xmx4g" )
library(xlsx)
library(lubridate)
library(gridBase)
library(grid)

#####
# wind barb function from http://stackoverflow.com/questions/32705013/plot-wind-barb-in-r

wind_barb <- function(x, mlength=0.1, wblength=0.025) {

# Calculate which / how many barbs
# any triangles (50)
fif <- floor(x /50)
# and then look for longer lines for remaining speed (10)
tn <- floor( (x - fif* 50)/10)
# and then look for shorter lines for remaining speed (5)
fv <- floor( (x - fif* 50 - tn* 10)/5)

# Spacing & barb length
yadj <- 0.5+mlength
dist <- (yadj-0.5) / 10
xadj <- 0.5+wblength
xfadj <- 0.5+wblength/2

# Create grobs
main_grob <- linesGrob(0.5, c(0.5, yadj ))

# 50 windspeed
if(fif != 0) {
  fifty <- c(yadj, yadj-dist*seq_len(2* fif) )
  fifx <- c(0.5, xadj)[rep(1:2, length=length(fifty))]
  fif_grob <- pathGrob(fifx, fifty, gp=gpar(fill="black"))
} else {
  fif_grob <- NULL
  fifty <- yadj+dist
}

# Ten windspeed
if(tn != 0) {
  tny <- lapply(seq_len(tn) , function(x) min(fifty) - dist*c(x, x-1))
  tn_grob <- do.call(gList,
    mapply(function(x,y)
      linesGrob(x=x, y=y, gp=gpar(fill="black")),
      x=list(c(0.5, xadj)), y=tny, SIMPLIFY=FALSE))
} else {
  tn_grob <- NULL
  tny <- fifty
}

# Five windspeed
if(fv != 0) {
  fvy <- lapply(seq_len(fv) , function(x) min(unlist(tny)) -dist* c(x, x-0.5))

```

```

fv_grob <- do.call(gList,
  mapply(function(x,y)
    linesGrob(x=x, y=y, gp=gpar(fill="black")),
    x=list(c(0.5, xfadj)), y=fvy, SIMPLIFY=FALSE))
} else {
  fv_grob <- NULL
}

# Draw
#grid.newpage()
grid.draw(gList(main_grob, fif_grob, tn_grob, fv_grob))
}
#####

files <- dir(path="C:\\Users\\ahawki03\\Desktop\\AQS_data\\", pattern="xlsx")
files <- files[grep("pdf", files, invert=TRUE)]
for (j in 1:length(files)) {
  wb <- loadWorkbook(paste("C:\\Users\\ahawki03\\Desktop\\AQS_data\\",files[j],sep=""))
  sheets <- getSheets(wb)
  i=6 #this is the NW site
  sheet <- sheets[[i]]
  name <- sheet$getSheetName()
  tmp <- read.xlsx2(paste("C:\\Users\\ahawki03\\Desktop\\AQS_data\\",files[j],sep=""), sheetName =
name,startRow=3,header=FALSE,stringsAsFactors=FALSE)
  tmp$X4 <- as.POSIXct((as.numeric(tmp$X4)-25569)*86400, tz="GMT", origin="1970-01-01")
  tmp$day <- day(ymd_hms(as.character(tmp$X4)))
  days <- na.omit(unique(day(ymd_hms(as.character(tmp$X4)))))

i=8 #this is the valley site
sheet <- sheets[[i]]
name <- sheet$getSheetName()
tmp1 <- read.xlsx2(paste("C:\\Users\\ahawki03\\Desktop\\AQS_data\\",files[j],sep=""), sheetName =
name,startRow=3,header=FALSE,stringsAsFactors=FALSE)
tmp1$X4 <- as.POSIXct((as.numeric(tmp1$X4)-25569)*86400, tz="GMT", origin="1970-01-01")
tmp1$day <- day(ymd_hms(as.character(tmp1$X4)))

i=10 #this is the valley 5 minute met data; X4 and X6 are the scalar WS and WD
sheet <- sheets[[i]]
name <- sheet$getSheetName()
met <- read.xlsx2(paste("C:\\Users\\ahawki03\\Desktop\\AQS_data\\",files[j],sep=""), sheetName =
name,startRow=3,header=FALSE,stringsAsFactors=FALSE)
met$X1 <- as.POSIXct((as.numeric(met$X1)-25569)*86400, tz="GMT", origin="1970-01-01")
#met <- met[-(1:2),] #take out the units
met$day <- day(ymd_hms(as.character(met$X1)))

pdf(file=paste("C:\\Users\\ahawki03\\Desktop\\AQS_data\\",files[j], ".pdf",sep=""),width=11,height=8.5)
par(mfrow=c(2,1))
for (i in 1:length(days)) {
  plot(as.numeric(as.character(tmp$X5[tmp$day==days[i]]))~tmp$X4[tmp$day==days[i]],xaxt='n',log="y",ylim=c(1,120
),pch=20,ylab="ppb",xlab="time",main=paste("Labadie",unique(strptime(na.omit(tmp$X4[tmp$day==days[i]]),format=
"%Y-%m-%d"))))
  r <- as.POSIXct(round(range(na.omit(tmp$X4[tmp$day==days[i]])), "hours"))
  axis.POSIXct(1, at = seq(r[1], r[2], by = "hour"), format = "%H")
}

```

```

points(as.numeric(as.character(tmp1$X5[tmp1$day==days[i]]))~tmp1$X4[tmp1$day==days[i]],col="red",pch=2)
legend("topleft", c("NW","Valley"),pch=c(20,2),col=c("black","red"))
met1 <- na.omit(met[met$day==days[i],])
met1$X4 <- as.numeric(met1$X4)
met1$X6 <- as.numeric(met1$X6)
met1$X16 <- as.numeric(met1$X16)/10
met1 <- na.omit(met1)
points(met1$X16~met1$X1,pch=20,col="green")
plot(as.numeric(as.character(tmp$X5[tmp$day==days[i]]))~tmp$X4[tmp$day==days[i]],xaxt='n',yaxt='n',xlab="",ylab=
"",ylim=c(80,120))
if (length(met1$X1)>0) {
vps <- baseViewports()
pushViewport(vps$inner, vps$figure, vps$plot)
# Plot
for (i in 1:nrow((met1))) {
  pushViewport(viewport(
    x=unit(met1$X1[i], "native"),
    y=unit(as.numeric(met1$X1[i])*0+100, "native"),
    angle=360-(as.numeric(met1$X6[i])))
    wind_barb((as.numeric(met1$X4[i])*10))
  popViewport()
}

popViewport(3)
}#end the length check of met1

}
dev.off()
}

bext_24
outdvis

```